

CLAIMS

1. (Currently Amended) A computer-readable storage medium apparatus having computer-executable instructions for performing ephemeral garbage collection, the computer-readable storage medium apparatus being accessible by a computing device, the instructions comprising:

requesting a list from a tracking mechanism, wherein the tracking mechanism comprises a write-watch mechanism, and the list:

~~identifying~~ identifies a plurality of memory locations that have been accessed since a last ephemeral garbage collection process, each memory location corresponding to one of a plurality of cards that are associated with ~~one or more~~ objects allocated from within a memory heap, each of the plurality of cards being associated with a card table, wherein the card table identifies one or more of the plurality of cards with ~~one or more~~ objects that have been accessed;

comprises a bitmap and each bit within the bitmap corresponds to one of the plurality of cards, modification of the bitmap occurring when a corresponding bit is set at the time that the card is trimmed to disk;

creating, during the ephemeral garbage collection process, a bundle table containing entries identifying a plurality of bundles, wherein each of the plurality of bundles identifies groupings of subsets of the plurality of cards;

marking, during the ephemeral garbage collection process, two or more of the plurality of bundles identified in the bundle table using the list, wherein the marked

bundles identify groupings of subsets of the plurality of marked cards having associated objects that have been accessed since the last garbage collection process;

for each marked bundle identified in the bundle table, determining at least one marked card in a grouping of subsets of the plurality of marked cards identified by the marked bundle;

for each determined marked card, determining at least one accessed object associated with the marked card; and

performing garbage collection upon the at least one accessed object.

2. (Canceled)

3. (Currently Amended) The computer-readable storage medium apparatus of claim [[2]] 1, wherein the write-watch mechanism operates within a memory manager.

4. (Currently Amended) The computer-readable storage medium apparatus of claim [[2]] 1, wherein the write-watch mechanism records a first access to one of the plurality of memory locations.

5. (Canceled)

6. (Currently Amended) The computer-readable storage medium apparatus of claim [[2]] 1, wherein the list of memory locations is maintained in response to a request from the ephemeral garbage collection process.

7. (Previously Presented) The computer-readable storage medium apparatus of claim 1, further comprising resetting the list of memory locations.

8. (Previously Presented) The computer-readable storage medium apparatus of claim 1, wherein the subset of cards corresponds to a number of cards that are tracked using a page of memory storing the card table.

9. (Previously Presented) The computer-readable storage medium apparatus of claim 1, wherein identifying the marked bundle comprises marking a bit associated with the marked bundle table within a bundle bitmap based on the memory locations within the list.

10. (Previously Presented) The computer-readable storage medium apparatus of claim 9, wherein marking the bit comprises setting the bit.

11. (Previously Presented) The computer-readable storage medium apparatus of claim 1, wherein determining the at least one marked card comprises scanning a card bitmap having a bit for each of the plurality of cards, the bit for each marked card being different than another bit associated with one of the cards that was not accessed.

12. (Currently Amended) A method for executing statements within a program to support ephemeral garbage collection, the method comprising:

specifying a range of card table memory to watch during program execution by calling a write-watch mechanism that performs tracking of the access to the card table

memory and maintains a write-watch list that identifies cards accessed within the card table memory since a garbage collection process was last performed, , each card being associated with and updated upon access to ~~one or more~~ objects allocated within a memory heap, the memory heap being divided into the plurality of cards with each card being grouped into one of a plurality of bundles, wherein one of the plurality of bundles corresponds to a subset of that plurality of cards that are tracked using a page of card table memory;

storing a value within the memory heap at a memory location specified by a storing statement,

marking one of the plurality of cards within the card table memory corresponding to the memory location;

tracking access to the card table memory by the write-watch ~~write-watch~~ mechanism;

creating, only during an ephemeral garbage collection process, one or more bundle tables containing entries identifying groupings of the cards in the plurality of bundles, for each stored statement within the program, the ephemeral garbage collection process occurring exclusive of the program execution process;

updating, during the ephemeral garbage collection process, at least one bundle table by marking the entries in the bundle table based on information obtained from the write-watch list, wherein the updated marked bundle table identifies groupings the plurality of marked cards having associated objects that have been accessed since a last garbage collection process;

for each marked bundle table, determining during the ephemeral garbage collection process at least one marked card in a grouping of the plurality of marked cards identified by the marked bundle table;

for each marked card, determining during the ephemeral garbage collection process, at least one accessed object associated with the marked card; and

performing garbage collection during the ephemeral garbage collection process upon the at least one accessed object.

13. (Canceled)

14. (Previously Presented) The method of claim 12, wherein the write-watch mechanism resides within a memory manager and sets bits in the card table memory upon access to at least one of the plurality of cards.

15. and 16. (Canceled)

17. (Previously Presented) The method of claim 12, wherein an ephemeral garbage collection process requests the write-watch list when performing a garbage collection cycle.

18. (Previously Presented) The method of claim 12, wherein the ephemeral garbage collection process determines a marked bundle based on the write-watch list.

19. (Currently Amended) A memory management system, the system comprising:

a processor; and

a memory into which a plurality of instructions are loaded and into which a plurality of objects are dynamically allocated, the memory having a heap into which the objects are allocated, the heap being divided into a plurality of cards which are grouped into a plurality of bundles, each card being associated with ~~one or more~~ a group of the plurality of objects[[:]], wherein upon execution of the plurality of instructions by the processor, the system being configured to:

request a list from a tracking mechanism, the list ~~only~~ identifying memory locations that have been written into since a last garbage collection cycle, each memory location corresponding to one of the plurality of cards associated with a card table, wherein the card table identifies one or more cards that have been accessed, the card table and cards being marked to identify the one or more of the plurality of cards with the group of ~~one or more~~ objects that have been accessed, during a first program execution process exclusive of an ephemeral garbage collection process;

creating, during the ephemeral garbage collection process, one or more bundle tables wherein each bundle table identifies groupings of the plurality of cards in the plurality of bundles;

updating, during the ephemeral garbage collection process, at least one bundle table by marking bundles within the bundle table based on the list, wherein the marked bundles corresponds to marked cards having associated objects that have been accessed since ~~[[a]]~~ the last garbage collection cycle process;

determine, during the ephemeral garbage collection process, for each marked bundle within the bundle table, at least one marked card within the marked bundle, the at least one marked card indicating that ~~one or more~~ objects associated with the marked card have been accessed;

determine, during the ephemeral garbage collection process, for each marked card, the one or more objects that have been accessed; and

perform, during the ephemeral garbage collection process, garbage collection upon the one or more accessed objects.

20. (Previously Presented) The system of claim 19, wherein the tracking mechanism comprises a write-watch mechanism.

21. (Previously Presented) The system of claim 20, wherein the write-watch mechanism resides within a memory manager and sets bits in the card table upon access to at least one of the plurality of cards.

22. (Previously Presented) The system of claim 19, wherein the subset of cards corresponds to a number of cards that are tracked using a page of memory storing the card table.

23. (Previously Presented) The system of claim 19, wherein the marked bundle being identified by a marked bit associated with the marked bundle within a bundle bitmap based on the list.

24. (Currently Amended) The system of claim 19, further being configured to set comprising setting a bit in the card table to identify one or more cards that have been accessed at the time a card that has been accessed is trimmed to disk.

25. (New) The method of claim 12, wherein the write-watch mechanism sets bits in the card table memory upon access to at least one of the plurality of cards at the time that the card is trimmed to disk.

26. (New) The method of claim 12, further comprising:
determining whether the call to the write-watch mechanism resets a write-watch state by inquiring which cards have changed without being considered as having asked and thereby resetting the state, and

in an event the state is to be reset, placing a separate reset call to reset the range of card table memory without reporting whether the cards in the range have been marked.

27. (New) The system of claim 19, further being configured to:
determine whether the request resets a write-watch state by inquiring which memory locations have changed without being considered as having asked and thereby resetting the state, and

in an event the state is to be reset, the system places a separate reset request to reset a range of memory locations without reporting whether the memory locations in the range have been marked.